

### REMARKS

Claims 1, 4-14, 17 and 20-24 were pending at the time of examination. Claims ... have been amended. No new matter has been added. The Applicants respectfully request reconsideration based on the foregoing amendments and these remarks.

#### **Claim Rejections – 35 U.S.C. § 102**

Claims 1, 4-14, 17 and 20-24 were rejected under 35 U.S.C. 102(b) as being anticipated by Silberschatz et al., "Operating System Concepts – Module 21: The Unix System", <http://www.cse.ucsc.edu/~sbrandt/courses/Spring01/111/slides/mod21.1.pdf>, 1999 (hereinafter "Silberschatz"). The Applicants traverse the rejections.

Claim 1 recites the step of:

" modifying the running parent process associated with the defunct child process by creating an agent thread inside the running parent process, the agent thread being operable to force the running parent process to collect exit information for the defunct child process, thereby enabling the collection of exit information for the defunct child process associated with the running parent process without terminating the running parent process."

Thus, by creating an agent process inside the running parent process that forces the parent process to collect exit information for the defunct child process, it is possible to collect exit information for the defunct child process (i.e., reduce the number of zombie processes) without terminating the running parent process. It is respectfully submitted that Silberschatz does not show the creation of an agent inside a running parent process, which collects exit information for a defunct child process.

On the contrary, the cited sections of Silberschatz merely contain a general description of process control and process control calls, which is no different than the conventional techniques that are recited in the background section of the Applicants' specification. That is, slide 21.11 of Silberschatz states that "processes are identified by their process identifier, an integer", that there are parent and child processes and that parent processes may wait for a child process to terminate, and that a zombie process results when the parent of a defunct child process exists before the terminated child. Slide 21.12 merely shows conventional process control, and makes no mention of modifying a running parent process by attaching an agent thread, on the one hand, and using this agent thread to reap zombie processes or defunct processes, on the other hand.

These are two distinct ideas, neither of which are taught by Silberschatz, and both of which are merged in a non-obvious way by the techniques and mechanisms of the present invention.

Independent claims 9, 14, 17 and 20 recite features similar to the features of claim 1 and were rejected based on the same rationale. In light of the above remarks, the rejection of claims 1, 9, 14, 17, and 20 under 35 U.S.C. 102(b) is unsupported by the art and should be withdrawn. Similarly, the respective dependent claims 4-13 and 21-24 are neither anticipated nor rendered obvious for at least the reasons noted above.

### Conclusion

The Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,  
BEYER WEAVER & THOMAS, LLP



Fredrik Mollborn  
Reg. No. 48,587

P.O. Box 70250  
Oakland, CA 94612-0250  
(650) 961-8300